REMARKS

Reconsideration of the present application is respectfully requested. Claims 1-24 and 37-39 were previously canceled. In this amendment, claim 40 has been amended. No claims have been canceled or added in this amendment. No new matter has been added.

Therefore, claims 25-35 and 40-49 are now pending.

Request for Telephone Interview

Applicants respectfully request that the Examiner contact the undersigned at (408) 720-8300 to schedule a telephone interview, in the event the Examiner does not find the present application to be allowable after considering this response.

Prior Art Rejections

Claims 25, 26, 28-31, 35, 43, 44, and 46-48 were rejected under 35 U.S.C. § 103(a) based on U.S. Patent no. 6,434,681 of Armangau et al. ("Armangau") in view of U.S. Patent no. 6,175,900 of Forin et al. ("Forin"). Claims 32-34 were rejected under 35 U.S.C. § 103(a) based on Armangau in view of U.S. Patent no. 6,665,815 of Goldstein et al. ("Goldstein").

Applicants respectfully traverse the rejections. As with the previous amendment, the above amendments are submitted only to place the claims in what Applicants consider to be better form. The amendments are *not* made in response to the rejections or to comply with any statutory requirement of patentability, since no such amendments are believed to be necessary (as will be apparent from the discussion below).

Independent claims 25, 35 and 43

Claim 43 recites (as amended):

> 43. (Currently amended) A method comprising:
> maintaining an active map of information indicating in-use blocks and free blocks associated with a file system;
> maintaining a plurality of persistent point-in-time images, each persistent point-in-time image representing a state of said file system at a particular point in time; and
> **generating a summary map as a logical union of active maps included in at least two of said persistent point-in-time images**.
> (Emphasis added.)

Claim 43 stands rejected under 35 U.S.C. § 103(a) based on Armangau in view of Forin. To support a rejection for obviousness, the cited combination of references must teach or suggest *all of the claim limitations*. *In re Vaeck*, 947 F.2d 488, 20 USPQ.2d 1438 (Fed. Cir. 1991); MPEP § 706.02(j) (emphasis added). Applicants respectfully submit that no combination of Armangau and/or Forin discloses *all of the limitations* of claim 43 or renders the claimed method obvious.

The Examiner admits that Armangau fails to disclose "as a logical union" (Final Office Action, p. 3). However, the Examiner contends that Forin teaches this feature, and that it would be obvious to combine the teachings of Forin and Armangau to achieve the present invention. Applicants respectfully disagree.

Applicants respectfully maintain the arguments set forth on pages 9-14 of their response filed on 10/18/2006, and those arguments are incorporated by reference into this response. In an effort avoid unnecessary verbosity, Applicants will not expressly repeat those arguments here.

Applicants will now respond to the Examiner's comments in the Advisory Action.

Response to Advisory Action

(a). Neither Armangau nor Forin discloses or suggests *generating a summary map as a logical union of active maps included in at least two of said persistent point-in-time images*.

(1). No logical union operation disclosed

Applicants appreciate the Examiner's detailed clarification of her rationale, in the Advisory Action. Applicants respectfully submit, however, that the Examiner's reasoning is flawed in relation to her interpretation of Forin.

Applicants have argued, and herein maintain, that assuming *arguendo* Forin discloses using any logic operation to form a summary map, it could only be a logical *intersection* operation, not a logical *union* operation as recited in claim 43 (see pages 10-11 of Applicants' last response). The Examiner disagrees with that assertion and explains her reasoning on pages 2-5 of the Advisory Action. The Examiner correctly notes that Forin discloses four possible states for a memory block, which the Examiner calls "State 1", "State 2", "State 3", and "State 4", corresponding to bit combinations "01, "11", "10" and "00", respectively. However, the Examiner states, "Of all four states above, only two states: State 1 (free blocks) and State 2 (in-use blocks), are mentioned in the final office action" (Advisory Action, page 3). Thus, the Examiner **ignores** half of the possible input states in attempting to support her conclusion, which has led to a significant error in the Examiner's reasoning. The Examiner has focused only on "State

1" and "State 2", ignoring what would happen if the inputs to the logic operation include "State 3" and/or "State 4". One cannot understand or correctly characterize a logic operation by looking at only some of the input states that can affect the output.

Applicants will explain this point further below. First, however, Applicants would like to address the Examiner's following statement from the Advisory Action (p. 3):

> It is brought to Applicant's attention that the condition clause "if both blocks are free" can be interpreted as: **if block A is free "*and*" block B is free, then Block C is free** (wherein block C is a single piece of memory in the coalescence of two adjacent blocks).
>
> Note that Applicants interpreted the term "*and*" in the sentence above as a **logical AND operation**, however, **"and" is not a logical AND operation**, rather, **"*and*" is a conjunction as presented in that sentence,** whereas the **logical AND operation** is a logical operation applies to binary numbers are bit values, which consist of 0, and 1. Advisory action, page 3 (emphasis original)

Applicants fully understand the distinction between the conjunction "*and*", which is a linguistic element, and the *logic AND* operation, which applies to binary numbers. Applicants respectfully submit, however, that the linguistic elements "*and*" and "*or*" do in fact represent basic conceptual/linguistic "logic", which by definition can be represented as binary logic operations, and likewise, binary logic operations can be explained or analyzed linguistically. That is precisely what Applicants did in their last response (p. 10), where they stated, "In Forin, the only situation in which two adjacent blocks should be coalesced into a single piece of memory is if *both* blocks are free, i.e., *both* of their bitmap values are '01'. That is analogous to a logic intersection (*AND*) operation, *not* a logical union (OR)." Indeed, the Examiner has attempted to do this same sort of linguistic-to-binary 'translation' on page 4 of the Advisory Action, thus implicitly recognizing that there is a correspondence between conceptual/linguistic logic and

binary logic. However, the Examiner's analysis is incorrect, as will now be further explained.

In the Advisory Action (pp. 4-5), the Examiner states:

First, the condition clause **"if both blocks are free", then block C is free**; implies for possible outcomes as:

Outcome 1: if block A is ***free***, block B is ***free***, then block C is ***free***.

Outcome 2: if block A is ***free***, block B is ***in-used***, then block C is ***in-used***.

Outcome 3: if block A is ***in-used***, block B is ***free***, then block C is ***in-used***.

Outcome 4: if block A is ***in-used***, block B is ***in-used***, then block C is ***in-used***.

Second, the following outcomes show the results when applying the two bit values of Forin (**01** corresponds to **free**, and **11** corresponds to **in-used**, See State 1, 2 above) to the corresponding allocation states:

Outcome 1: if block A is ***01***, block B is ***01***, then block C is ***01***.

Outcome 2: if block A is ***01***, block B is ***11***, then block C is ***11***.

Outcome 3: if block A is ***11***, block B is ***01***, then block C is ***11***.

Outcome 4: if block A is ***11***, block B is ***11***, then block C is ***11***.

Third, applying a logical AND operation to the above outcomes, the outcomes could be written as:

Outcome 1: 01 AND 01 = 01 (correct)

Outcome 2: 01 AND 11 = 11 (***incorrect***)

Outcome 3: 11 AND 01 = 11 (***incorrect***)

Outcome 4: 11 AND 11 = 11 (correct)

for the above reasons, it is incorrect when applying an intersection (AND) operation to the outcome two, and outcome three, force, and applying a logical OR operation to the above outcomes, the outcomes can be written as:

Outcome 1: 01 OR 01 = 01 (correct)

Outcome 2: 01 OR 11 = 11 (correct)

Outcome 3: 11 OR 01 = 11 (correct)

Outcome 4: 11 OR 11 = 11 (correct)

Examiner thus maintains the rejection, and asserts that foreign does teach a logical union (OR).

In response, Applicants submit that, in order to correctly determine what type of logic operation Forin discloses in connection with coalescing adjacent blocks, one must

consider **all of the possible input states**. The Examiner has focused only on "State 1" (01) and "State 2" (11), **ignoring what would happen if the inputs to the logic operation include "State 3" (10) or "State 4" (00)**. By ignoring two of the four possible bit combinations disclosed in Forin as possible inputs (i.e., 10 and 00), the Examiner has mischaracterized the logic operation. For example, performing a (bitwise) logical OR (union) on the pair of inputs [00, 01] (a case which the Examiner ignored) produces a result of 01; that result would be **incorrect** in the system of Forin, however, since "00" in Forin represents a sub-allocated block (col. 7, lines 37-43), which should not be coalesced with a free block.

Furthermore, the Examiner's logic on pages 4-5 of the Advisory Action improperly relies upon applying **bitwise** (bit-by-bit) logic to the arbitrary bit combinations that Forin uses to represent the possible input states (and again, improperly ignoring two of the four possible input states). By performing **bitwise** logic, the Examiner loses sight of the **overall concept** that Forin discloses for coalescing, and thereby mischaracterizes the logic function. To determine what logic operation Forin discloses, one must understand and correctly represent the **overall concept**, taking into consideration **all possible input states**. Applicants will now do that, as follows:

As Applicants have noted, the **only** situation disclosed in Forin in which two adjacent blocks A and B should be coalesced into a single piece of memory is when **both** blocks are **free**. Thus, there are only two input conditions that matter for purposes of determining whether two blocks A and B can be coalesced into a single piece of memory: a block is **free** or it is **not free**. Although Forin does not define a state called "not free", **for purposes of analyzing the logic operation** we can properly represent

all three bit combinations "00", "10" and "11" in Forin as a single logic state, **not free**, since those three states are treated the same way in Forin when determining whether to coalesce two adjacent blocks. Therefore, for purposes of discussion, we can logically represent the bit combination "01" in Forin as a **logic 1** input (meaning "**free**") and collectively represent **all** of bit combinations "00", "10" and "11" in Forin as a **logic 0** input (meaning "**not free**"). Hence, for purposes of determining whether to coalesce two blocks A and B, blocks A and B each can have possible input values of logic **1** (free) or logic **0** (not free). Again, while Applicants have combined/generalized three of the four possible input states into one input state (not free), those three states are nonetheless fully represented in the logic function described below and are not ignored (as the Examiner did), since those three states are all properly treated the same way for purposes of coalescing.

On the output side of the logic function, there are two possible outcomes: a decision to **coalesce** or **do not coalesce**. Accordingly, we can represent a decision to coalesce as a logic 1 in the output and a decision not to coalesce as a logic 0 in the output. Thus, the determination of whether two blocks A and B should be coalesced in Forin is **correctly** represented by the following truth table, where a logic 1 in the input means **free** (bit combination 01 in Forin), a logic 0 in the input means **not free** (*any* of bit combinations 10, 00, or 11 in Forin), a logic 1 in the output means that blocks A and B **should** be coalesced, and a logic 0 in the output means that blocks A and B **should not** be coalesced:

A	B	Output (Coalesce?)
0	0	0
0	1	0

1	0	0
1	1	1

It is clear that the above truth table represents a logical **AND** (***intersection***) operation, **not** a logic OR (union) operation as recited in Applicants' claims. Therefore, Applicants maintain that, if Forin discloses using any logic operation to form a "summary map", it is a logical **intersection** operation, not a logical **union** operation as recited in Applicants' claims.

(2) Assuming *arguendo* Forin discloses a logical union operation, the cited art still fails to disclose or suggest *generating a summary map as* a logical union of active maps.

It is incorrect to argue that the separate limitations of a claim taken by themselves are known or obvious and that "therefore" the whole subject-matter claimed is obvious. Under section 103, the claimed invention **as a whole** must be obvious to support a rejection. The Examiner has acknowledged that Armangau does not disclose "as a logical union". Assuming *arguendo* that Forin discloses "as a logical union", it does not relate to generating a summary map based on active maps.

Further, there is simply **no motivation or suggestion** for one of ordinary skill in the art to make such a combination, nor is there any **reasonable expectation of success**. The motivation alleged by the Examiner to combine Forin with Armangau does not withstand scrutiny. According to the Examiner, the motivation would be, "in order to represent[] an allocation state of the corresponding memory in view of Forin, as doing so would give the added benefit of managing the memory blocks and tracks

changes in allocation state via hierarchical bit map as taught by Forin" (Final Office Action, p. 7). The Examiner's statement is, essentially, that the motivation to combine the teachings of Forin with those of Armangau would be: to apply the principles of Forin to the system of Armangau. That is circular (i.e., flawed) logic. That is exactly the type of conclusory, unsupported statement of motivation which the courts have held to be inadequate to sustain an obviousness rejection. The Examiner must show **why** combining the teachings of the references in this way would be desirable. "[T]he showing [of motivation to combine the references] **must be clear and particular**" findings of fact based on **actual evidence**, not merely broad conclusive statements. *In re Dembiczak*, 175 F.3d 994, 999 (Fed. Cir. 1999)(emphasis added). Furthermore, it is not even clear that a "hierarchical bit map as taught by Forin" could even be successfully implemented in the system of Armangau. As such, there is **no reasonable expectation of success**.

Thus, even assuming *arguendo* Forin discloses a logical union operation, the cited art still fails to disclose or suggest **generating a summary map *as*** a logical union of active maps.

Hence, neither Armangau nor Forin discloses or suggests generating a summary map *as a logical union of* active maps included in at least two persistent point-in-time images. For at least the above reasons, therefore, the rejections are improper and should be withdrawn.

(b). Armangau does not teach "maintaining an active map of information indicating in-use blocks and free blocks associated with a file system" as alleged

by the Examiner.

Applicants previously argued this point on pages 11-12 of their last response. The Examiner responds on page 6 of the Advisory Action.

Applicants respectfully maintain that Armangau does not disclose or suggest maintaining an active map of information indicating in-use blocks **and free blocks associated with a file system** per claims 25, 33 and 43, much less an **active** file system per independent claims 25 and 32. The Examiner contends that "Armangau teaches an active map of information indicating free blocks *as a list of pointers to free tracks*" (col. 13, line 66 to col. 14, line 7). The "list of pointers to free tracks" to which the Examiner refers are elements 109 and 110 in Fig. 5 of Armangau. Elements 109 and 110 are associated with **snapshots**, **not** an **active file system**. On the other hand, at least some of elements 105, 106, 107, and 108 do arguably relate to an active file system; however, those elements do **not** contain any information about **free blocks**, and therefore, they do not contain sufficient information to constitute a map of in-use **and free blocks** of an **active** file system.

Furthermore, the **separate** sets of information represented by elements 105 – 110 in Armangau cannot reasonably be viewed collectively as a "map" of in-use and free blocks of a file system. Nowhere does Armangau does any **map** of information indicating both **in-use blocks and free blocks** associated with a particular file system, per claims 25, 32 and 43, much less an **active** file system as recited in claims 25 and 32.

(c). Armangau does not teach or suggest "generating a summary map as a

logical union of active maps included in at least two of said persistent point-in-time images".

Applicants previously argued this point on page 12 of their last response. The Examiner responds on page 7 of the Advisory Action. First, the Examiner states:

> In contrast, under similar rationale as provided in (b), the same reasoning would be applicable to "generating a summary map of active maps included in at least two of said persistent point in time images". It is thus clearly shown by Armangau active maps as explained in (b). (Advisory Action, p. 7)

Applicants respectfully submit that the Examiner's above-quoted statement is conclusory and incorrect. The Examiner improperly contends that two different claim limitations are satisfied by the same disclosure in Forin, without explaining how that disclosure is relevant to the second limitation. The Examiner thereby improperly glosses over the differences in these claim limitations. For at least this reason, the Examiner has not set forth a *prima facie* case of obviousness. Applicant respectfully submits that, if the Examiner maintains this rejection or asserts a similar new rejection, the Examiner is obligated to clarify **exactly what feature in Armangau** the Examiner considers to be a summary map of active maps included in at least two of said persistent point-in-time images, as recited in Applicants' claims. "It is important for an examiner to properly communicate the basis for a rejection so that the issues can be identified early and the applicant can be given a fair opportunity to reply." MPEP § 706.02(j).

The Examiner then goes on to state:

> Further, Fig. 6 of Armangau has shown a summary map as "List of pointers to track in snapshot disks for production volume extent", which

include at least two of said persistent point-in-time (*i.e. a date/time stamp when the snapshot copy was made, col. 18, lines 18-37*) images as "snapshot track 0 and snapshot track 1" (**See Fig. 6**). (Advisory Action, p. 7, emphasis added.)

In response, Applicants submit that Fig. 6 does **not** indicate any **free** blocks, and certainly does **not** indicate **free** blocks of an **active** file system as per claims 25 and 32. As discussed above, that is an explicitly recited characteristic of the "active maps" in Applicants' claims. Consequently, Applicants maintain that Armangau does not teach or suggest "generating a summary map as a logical union **of active maps** included in at least two of said persistent point-in-time images."

In view of the above points, the cited *combination* of references does not disclose or suggest *all of the limitations* of Applicants' independent claims. Therefore, a *prima facie* case of obviousness has not been established.

(d) Claim 32: The cited art also fails to disclose or suggest "deleting said particular snapshot, *wherein said deleting involves, for a block used by said particular snapshot, indicating said block is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot.*"

Applicants previously argued this point on pages 13-14 of their last response. The Examiner responds on pages 7-8 of the Advisory Action. First, Applicants note that the Examiner misstates Applicants' argument as "Applicants argued that Goldstein does not disclose the leading a snapshot." That is **not** what Applicants argued -- see page 13 of Applicants' last response. Applicants argument on this issue is set forth in the

above heading "(d)" and below.

The Examiner contends that Goldstein discloses the above-mentioned functionality of claim 32 at col. 4, lines 41-51, and Fig. 4. In this regard, Applicants note that the Examiner has merely repeated the allegation and citation from the Final Office Action without attempting to rebut Applicants' argument (set forth on pages 13-14 of Applicants' last response).

As Applicants noted, Goldstein **does** disclose deleting a snapshot. However, Applicants find absolutely no disclosure, or even a hint, anywhere in Goldstein, that deleting a snapshot may involve, for a block used by said particular snapshot, **indicating said block is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot**, as recited in claim 32. For this additional reason, therefore, claim 32 and all claims which depend on it are patentable over the cited art.

(e) Conclusion

For each of the above independent reasons, the rejections are improper and should be withdrawn.

Dependent Claims

In view of the above remarks, a specific discussion of the dependent claims is considered to be unnecessary. Therefore, Applicants' silence regarding any dependent claim is not to be interpreted as agreement with, or acquiescence to, the rejection of such claim or as waiving any argument regarding that claim.

Conclusion

For the foregoing reasons, the present application is believed to be in condition for allowance, and such action is earnestly requested.

If there are any additional charges/credits, please charge/credit our deposit account no. 02-2666.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: 11/16/2006

/Jordan M. Becker/
Jordan M. Becker
Reg. No. 39,602

Customer No. 48102
12400 Wilshire Blvd.
Seventh Floor
Los Angeles, CA 90025
(408) 720-8300